Nokia Docket Number: NC17525
The Patent Application Papers of: KARCZEWICZ, et al.

VIDEO DECODER ARCHITECTURE AND METHOD FOR USING SAME

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is related to and claims priority from Provisional Application Number 60/259.529 filed on January 3, 2001, incorporated herein by reference.

BACKGROUND

[0002] This invention relates generally to the field of the multimedia applications. More particularly, this invention relates to a decoder/decompressor and method for decoding streaming video.

[0003] Multimedia applications that include audio and streaming video information have come into greater use. Several multimedia groups have established and proposed standards for compressing/encoding and decompressing/decoding the audio and video information. MPEG standards, established by the Motion Picture Expert Group, are the most widely accepted international standards in the field of the multimedia applications. ITU-Telecommunications Standardization have developed video coding standards established by the Video Coding Experts Group (VCEG). Other standards are JPEG and Motion JPEG established by the Joint Photographic Expert Group.

[0004] The following are incorporated herein by reference:

[0005] Gisle Bjontegaard, "H.26L Test Model Long Term Number 5 (TML-5) draft0", document Q15-K-59, ITU-T Video Coding Experts Group (Question 15) Meeting, Oregon, USA 22-25 August, 2000. Keiichi Hibi, " Report of the Ad Hoc Committee on H.26L Development", document Q15-H-07, ITU-T Video Coding Experts Group (Question 15) Meeting, Berlin, 03-06 August, 1999. Gary S. Greenbaum, "Remarks on the H.26L Project: Streaming Video Requirements for Next Generation Video Compression Standards", document Q15-G-11, ITU-T Video Coding Experts Group (Question 15) Meeting, Monterey, 16-19 February, 1999. G. Bjontegaard, " Recommended Simulation Conditions for

H.26L", document Q15-I-62, ITU-T Video Coding Experts Group (Question 15) Meeting, Red Bank, New Jersey, 19-22 October, 1999. G. Bjontegaard, "H.26L Test Model Long Term Number 6 (TML-6) draft0", document VCEG-L45, ITU-T Video Coding Experts Group Meeting, Eibsee, Germany, 09-12 January 2001. ATM & MPEG-2 Integrating Digital Video into Broadband Networks by Michael Orzessek and Peter Sommer (Prentice Hall Upper Saddle River New Jersey).

[0006] The purpose of the video coding is to remove the redundancy in the image sequence so that the encoded data rate is commensurate with the available bandwidth to transport the video sequence while keeping the distortion between the original and reconstructed images as small as possible. The redundancy in video sequences can be categorized into spatial and temporal redundancy. Spatial redundancy refers to the correlation between neighboring pixels in a frame while temporal redundancy refers to correlation between neighboring frames.

[0007] For every pixel of a image, color information must be provided. Typically, color information is coded in terms of the primary color components red, green and blue (RGB) or using a related luminance/chrominance model, known as the YUV model.

[0008] Typical video codec employs three types of frames: intra frames (I-frames) and predicted frames (P-frames) and Bi-directional-frame (B-frames). Coding of a frame is performed independently from the others. I-frame, exploits only the spatial correlation of the pixels within the frame. Coding of P-frames exploits spatial as well temporal redundancies between the successive frames. Since in a typical video sequence the objects appearing in a sequence don't change rapidly from one frame to the next frame, i.e., the adjacent frames in a sequence are highly correlated, higher compression efficiencies are achieved when using P-frames. The terms frame and picture have been interchanged in the art. A frame contains all the color and brightness information that is need to display a picture. A picture is divided into a number of blocks, which are grouped into macroblocks. Each block contains a number of lines, with each

line holding a number of samples of luminace or chrominance pixel values from a frame.

[0009] FIG.s 1 and 2 show multimedia coding using MPEG as an example. FIG. 1A is a diagram of an MPEG audio and video decoder 120 that performs decompression of the video and/or audio data which has been compressed and coded according to the MPEG algorithm. The system decoder 110 reads the encoded MPEG data stream 101 having interlaced compressed video and/or audio data, and generates necessary timing information; Video Presentation Time Stamp (VPTS) 104; System Clock Reference (SCR) 105 which is also referred to as system time clock (STC); Audio Presentation Time Stamp (APTS) 106; and separated video encoded bit streams 102 and audio encoded bit streams 103. The video decoder 111 decompresses the video data stream 102 and generates a decompressed video signal 107. The audio decoder 112 decompresses the audio data stream 103 and generates the decompressed audio signal 108. The decompressed video signal 107 is coupled to a display unit, while the decompressed audio signal 108 is coupled to an audio speaker or other audio generation means.

[0010] The MPEG encoded/compressed data stream may contain a plurality of encoded/compressed video data packets or blocks and a plurality of encoded/compressed audio data packets or blocks. An MPEG encoder encodes/compresses the video packets based on video frames, also referred to as pictures. These pictures or frames are source or reconstructed image data consisting of three rectangular matrices of multiple-bit numbers representing the luminance and chrominance signals. For example, H.263+ uses four luminance blocks and two chrominance blocks of 8X8 pixels each. FIGS. 2A-2C illustrate the type of encoded/compressed video frames that are commonly utilized for MPEG standard. FIG. 2A depicts an Intra-frame or I-type frame 200. The I-type frame or picture is a frame of video data that is coded without using information from the past or the future and is utilized as the basis for decoding/decompression of other type frames. FIG. 2B is a representation of a Predictive-frame or P-type frame 210. The P-type frame or picture is a frame that is encoded/compressed using motion compensated prediction from an I-

type or P-type frame of its past, in this case, I.sub.1 200. That is, a previous frame is used to encode/compress a present given frame of video data. 205a represents the motion compensated prediction information to create a P-type frame 210. FIG. 2C depicts a Bi-directional-frame or B-type of frame 220. The B-type frame or picture is a frame that is encoded/compressed using a motion compensated prediction derived from the I-type reference frame (200 in this example) or P-type reference frame in its past and the I-type reference frame or P-type reference frame (210 in this example) in its future or a combination of both. B-type frames are usually inserted between I-type frames or P-type frames. FIG. 2D represents a group of pictures in what is called display order I.sub.1 B.sub.2 B.sub.3 P.sub.4 B.sub.5 P.sub.6. FIG. 2D illustrates the B-type frames inserted between I-type and P-type frames and the direction which motion compensation information flows.

[0011] Motion compensation refers to using motion vectors from one frame to the next to improve the efficiency of predicting pixel values for encoding/compression and decoding/decompression. The method of prediction uses the motion vectors to provide offset values and error data that refer to a past or a future frame of video data having decoded pixel values that may be used with the error data to compress/encode or decompress/decode a given frame of video data.

[0012] The capability to decode/decompress P-type frames requires the availability of the previous I-type or P-type reference frame and the B-type frame requires the availability of the subsequent I-type or P-type reference frame. For example, consider the encoded/compressed data stream to have the following frame sequence or display order:

[0013] I.sub.1 B.sub.2 B.sub.3 P.sub.4 B.sub.5 P.sub.6 B.sub.7 P.sub.8 B.sub.9 B.sub.10 P.sub.11 . . . P.sub.n-3 B.sub.n-2 P.sub.n-1 I.sub.n.

[0014] The decoding order for the given display order is:

[0015] I.sub.1 P.sub.4 B.sub.2 B.sub.3 P.sub.6 B.sub.5 P.sub.8 B.sub.7 P.sub.11 B.sub.9 B.sub.10 . . . P.sub.n-1 B.sub.n-2 I.sub.n.

[0016] The decoding order differs from the display order because the B-type frames need future I-type or P-type frames to be decoded. P-type frames require that the previous I-type reference frame be available. For example, P.sub.4 requires I.sub.1 to be decoded such that the encoded/compressed I.sub.1 frame needs to be available. Similarly, the frame P.sub.6 requires that P.sub.4 be available in order to decode/decompress frame P.sub.6. B-type frames, such as frame B.sub.3, require a past and future I-type or P-type reference frames, such as P.sub.4 and I.sub.1 in order to be decoded. B-type frames are inserted frames between I-type, P-type, or a combination during encoding and are not necessary for faithful reproduction of an image. The frames before an I-type frame, such as P.sub.n-1 in the example, are not needed to decode an I-type frame, and no future frames require P.sub.n-1 in order to be decoded/decompressed.

[0017] One problem with decoding is that the display process may be slower than the decoding process. For example, a 240.times.16 picture requires 3072 clock cycles to decode(76.8us at 40Mhz); it takes 200us to display 16 lines of video data at a 75 Hz refresh rate (13us.times.16=200us). The video frames are buffered before being displayed. There is usually a one frame delay between display and decoding. The difference between display and decoding leads to a condition known as tearing. Tearing occurs when the display frame is overwritten by the decoded frame.

[0018] FIG. 1B depicts tearing. A decoded/decompressed frame 132 of data representing the image of a closed door 133 is currently stored in a buffer 135. This decode/decompressed frame is currently being displayed on display unit 140. During this display period another decoded/decompressed frame 130 with data representing the image of an open door 131 is stored in buffer 135. The display unit 140 will now start displaying using information from the new frame now stored in 135. The result is a partial display of the first stored image 141 and partial display of the new stored image 142.

[0019] Video streaming has emerged as one of the essential applications over the fixed internet and- in the near future over 3G multimedia networks. In

streaming applications, the server starts streaming the pre-encoded video bitstream to the receiver upon a request from the receiver which plays the stream as it receives with a small delay or no delay. The problem with video streaming is that the best-effort nature of today's networks causes variations of the effective bandwidth available to a user due to the changing network conditions. The server should then scale the bitrate of the compressed video to accommodate these variations. In case of conversational services that are characterized by real-time encoding and point-to-point delivery, this is achieved by adjusting, on the fly, the source encoding parameters, such as quantization parameter or frame rate, based on the network feedback. In typical streaming scenarios when already encoded video bitstream is to be streamed to the client, the above solution can not be applied. A situation similar to tearing as described above would occur.

[0020] Thus, there is a need to provide bandwidth scalability which will allow a server to dynamically switch between the streams of encoded video in order to accommodate variations of the bandwidth available to the client.

[0021] The above-mentioned references are exemplary only and are not meant to be limiting in respect to the resources and/or technologies available to those skilled in the art.

SUMMARY

[0022] A new picture or frame type and method of using same is provided. This type of novel frame type is referred to as a SP-picture. The temporal redundancies are not exploited in I-frames, compression efficiency of I-frame coding is significantly lower than the predictive coding. The proposed method allows use of motion compensated predictive coding to exploit temporal redundancy in the sequence while still allowing perfect reconstruction of the frame using different reference frames. This new picture type provides for error resilience/recovery, bandwidth scalability, bitstream switching, processing scalability, random access and other functions.

[0023] The SP-type picture provides for, among other functions, switching between different bitstreams, random access, fast forward and fast error-recovery by replacing I-pictures to increase the coding efficiency. As will be demonstrated, SP-pictures have the property that identical SP-frames may be obtained even when they are predicted using different reference frames.

[0024] These and other features, aspects, and advantages of embodiments of the present invention will become apparent with reference to the following description in conjunction with the accompanying drawings. It is to be understood, however, that the drawings are designed solely for the purposes of illustration and not as a definition of the limits of the invention, for which reference should be made to the appended claims.

[0025] BRIEF DESCRIPTIONS OF THE DRAWINGS

[0026] FIG. 1A is a prior art block diagram of an MPEG decoding system.

[0027] FIG. 1B is a drawing showing the problem of tearing within prior art devices where frame data from two different frames stored consecutively in frame buffer memory is displayed on a display device.

[0028] FIG. 2A-2D are diagrams showing the prior art encoding/compression of video frames.

[0029] FIG. 3 is a block diagram of a generic motion-compensated predictive video coding system (encoder).

[0030] FIG. 4 is a block diagram of a generic motion-compensated predictive video coding system (decoder).

[0031] FIG. 5 is an illustration showing switching between bitstreams 1 and 2 using SP-pictures.

[0032] FIG. 6 is a block diagram of a decoder in accordance with the preferred embodiment of the invention.

[0033] FIG. 7 is an illustration of random access using SP-pictures.

[0034] FIG. 8 is an illustration of a fast-forward process using SP-pictures.

[0035] FIG. 9 is a set of graphs comparing coding efficiencies of I, P, and SP-pictures.

[0036] FIG. 10 is a set of graphs comparing performance of SP-pictures and I-pictures when used at fixed one second intervals.

[0037] FIG. 11 is a set of graphs comparing performance of SP-pictures and I-pictures in fast-forward application, one second Intervals.

[0038] DETAILED DESCRIPTION

[0039] The simplest way of achieving bandwidth scalability in case of pre-encoded sequences is by producing multiple and independent streams of different bandwidth and quality. The server dynamically switches between the streams to accommodate variations of the bandwidth available to the client.

[0040] Now assume that we have multiple bitstreams generated independently with different encoding parameters, such as quantization parameter, corresponding to the same video sequence. Since encoding parameters are different for each bitstream, the reconstructed frames of different bitstreams at the same time instant will not be the same. Therefore when switching between bitstreams, i.e., starting to decode a bitstream, at arbitrary locations would lead to visual artifacts due to the mismatch between the reference frames used to obtain predicted frame P(x, y). Furthermore, the visual artifacts will not only be confined to the switched frame but will further propagate in time due to motion compensated coding.

[0041] The main observation is that perfect (mismatch-free) switching between bitstreams, in the current standards, is possible only at the positions where the future frames/regions do not use any information previous to the current switching location and the information at the current location is made available to the client.

[0042] In prior art solutions, the approach adopted is to insert periodic I-frames during encoding and allow the switch to occur only at these I-frames. Since I-frames may be reconstructed independently from the previous frames, switching at these frames doesn't cause any mismatch. Thus a new type of picture or frame is needed as is architecture and methods of using said new frame type.

[0043] A new decoder architecture is provided which has the property that identical frames may be obtained even when they are predicted using different reference frames. The picture type obtained using this structure will be called SP-frame also may be referred to as picture.

[0044] A system for P-frame encoding and decoding is provided and is shown in FIG.s 3 and 4. Referring to FIG.s 3 and 4, a communication system comprising an encoder 300 of FIG. 3 and a decoder 400 of FIG. 4 is operable to communicate a multimedia sequence between a sequence generator and a sequence receiver. Other elements of the video sequence generator and receiver are not shown for the purposes of simplicity. The communication path between sequence generator and receiver may take various forms, including but not limited to a radio-link.

[0045] Encoder 300 is shown in FIG. 3 coupled to receive video input on line 301 in the form of a current frame to be encoded I(x, y), called the current frame. The video input may be provided to a motion estimation and coding block 370 through 305 and to an input of a subtractor 307. The motion estimation and coding block 370 may also be coupled frame memory 350 to receive indications of a previously coded and transmitted frame R(x, y), called a reference frame. The motion estimation and coding block 370 may also be coupled to multiplexor 380 to provide motion information for bitstream.

[0046] The current frame I(x, y), is partitioned into rectangular regions of M x N pixels. By (x, y) we denote location of the pixel within the frame. These blocks may be encoded using either only spatial correlation (intra-coded blocks) or both spatial and temporal correlation (inter-coded blocks). In what follows, we concentrate on inter blocks. Each of inter-coded blocks may be predicted from one of the previously coded and transmitted reference frame, which at given instant is available in the Frame Memory 350 of the encoder 300 and of the Frame Memory 450 of the decoder 400 in FIG. 4. The frame memory 350 is coupled to a Motion Compensated (MC) prediction block 360. The MC prediction block 360 is operable to generate a prediction frame P(x, y) which is provided to an input of subtractor 307 and adder 345. MC prediction block 360 is also coupled to the motion estimation and coding block 370 to receive motion information.

[0047] The prediction information may be represented by two dimensional motion vector (Δx, Δy) where Δx is the horizontal and Δy is the vertical displacement, respectively of the pixels between the current frame and the reference frame. The motion estimation and coding block 370 calculates the motion vectors (Δx, Δy). In the Motion Compensated (MC) Prediction block 360, the motion vectors together with the reference frame are used to construct prediction frame P(x, y):

$$P(x, y) = R(x+\Delta x, y+\Delta y).$$

[0048] Subsequently, the prediction error E(x, y), i.e., the difference between the current frame and the prediction frame P(x, y) is calculated by:

$$E(x, y)= I(x, y)- P(x, y).$$

[0049] In transform block 310, the prediction error for each K x L block is represented as weighted sum of a transform basis functions f.sub.ij(x, y),

$$E(X, Y) = \sum_{i=1}^{K} \sum_{J=1}^{L} C.sub.err (I,j) fsub.ij (X, Y).$$

[0050] The weights c.sub.err(I,j), corresponding to the basis functions are called transform coefficients. These coefficients are subsequently quantized in quantization block 320:

$$I.sub.err(I, j)=Q(C(I, j),QP)$$

[0051] where I.sub.err(i, j) are the quantized coefficients. The operation of quantization introduces loss of information -- the quantized coefficient can be represented with smaller number of bits. The level of compression (loss of information) is controlled by adjusting the value of the quantization parameter (QP).

[0052] The quantization block 320 is coupled to both a multiplexor 380 and an inverse quantization block 330 and in turn an inverse transform block 340. Blocks 330 and 340 provide prediction error which is added to the MC predicted frame P(x, y) by added 345 and the result stored in frame memory 350.

[0053] Motion vectors and quantized coefficients are further encoded using Variable Length Codes (VLC) which further reduce the number of bits needed for their representation. Encoded motion vectors and quantized coefficients as well as other additional information needed to represent each coded frame of the image sequence constitute a bitstream 415 which is transmitted to the decoder 400 of FIG. 4. Bitstream may be multiplexed 380 before transmission.

[0054] FIG. 4 shows the decoder 400 of the communication system. Bitstream 415 is received from encoder 300 of FIG. 3. Bitstream 415 is demultiplexed via demultiplexor 410 . Dequantized coefficients d.sub.err(I,j) are calculated in the inverse quantization block 420:

$$d.sub.err(i, j)=Q^{-1}(I.sub.err(I, j), QP).$$

[0055] In inverse transform block 430, the dequantized coefficients are used to obtain compressed prediction error:

$$E.sub.c(X, Y) = \sum_{i=1}^{K} \sum_{J=1}^{L} C.sub.err (I, j) f.sub.ij (X, Y).$$

[0056] The pixels of the current coded frame are reconstructed by finding the prediction pixels in the reference frame R(x,y) using the received motion vectors and then adding to the compressed prediction error in adder 435 resulting in decoded video:

$$I.sub.c(x, y)= R(x+\Delta x, y+\Delta,y)+E.sub.c(x, y).$$

[0057] These values can be further normalized and filtered.

[0058] One of the key requirements for video streaming is to scale the transmission bitrate of the compressed video according to the changing network conditions. In case of conversational services that are characterized by real-time encoding and point-to-point delivery, this is achieved by adjusting on the fly the source encoding parameters, such as quantization parameter or frame rate based on the network feedback. In typical streaming scenarios when already encoded video sequence is to be streamed to the client the above solution can not be applied.

[0059] The simplest way of achieving bandwidth scalability in case of pre-encoded sequences is by producing multiple and independent streams of different bandwidth and quality. The server dynamically switches between the streams to accommodate variations of the bandwidth available to the client. Since the encoding algorithms employ motion-compensated prediction, switching between bitstreams at arbitrary P-type pictures, although possible, would lead to visual artifacts due to the mismatch between the reconstructed frames at the same time instant of different bitstreams. The visual artifacts will further propagate in time.

[0060] In the current video encoding standards, perfect (mismatch-free) switching between bitstreams is possible only at the positions where the future frames/regions do not use any information previous to the current switching location, i.e., at I-frames. Furthermore, by placing I-frames at fixed (e.g. 1 sec) intervals, VCR functionalities, such as random access or "Fast Forward" and "Fast Backward" (increased playback rate) for streaming video content, are achieved. User may skip a portion of video and restart playing at any I-frame location. Similarly, increased playback rate, i.e., fast-forwarding, can be achieved by transmitting only I–pictures.

[0061] It is, however, well known that I-frames require a lot more bits than the motion-compensated predicted frames. An embodiment of the present invention provides a novel picture type, SP-picture, which allows switching from one bitstream to another, enables VCR like functionalities without introducing any mismatch while still utilizing motion compensated prediction. One of the properties of SP-pictures is that identical SP-frames may be obtained even when different reference frames are used.

Bitstream switching:

[0062] An example of how to utilize SP-frames to switch between different bitstreams is illustrated in the FIG. 5. FIG. 5 shows two bitstreams corresponding to the same sequence encoded at different bitrates--bitstream 1 510 and bitstream 2 520. Within each encoded bitstream, SP-pictures may be placed at the locations at which one wants to allow switching from one bitstream to another (pictures S.sub.1 513 and S.sub.2 523). When switching from

bitstream 1 to bitstream 2, another SP-picture will be transmitted. This is show in FIG. 5 by picture S.sub.12 550. Pictures S.sub.2 523 and S.sub.12 550 are represented by different bitstreams, i.e., S.sub.2 (S.sub.12) uses the previously reconstructed frames from bitstream 2 as the reference frames, however their reconstructed values are identical.

Random Access:

[0063] Application of SP-pictures to enable random access is depicted in Figure 7. SP-pictures are placed at fixed intervals within bitstream 1 720 (e.g. picture S.sub.1 (730)) which is being streamed to the client. To each one of these SP-pictures there is a corresponding pair of pictures generated and stored as another bitstream (bitstream 2 (740)):

[0064] I-picture, I.sub.2 (750), at the temporal location preceding SP-picture.

[0065] SP-picture 710, S.sub.2, at the same temporal location as SP-picture.

[0066] Bitstream 1 (720) may then be accessed at a location corresponding to an I-picture in bitstream 2 (740). For example to access bitstream 1 at frame I.sub.2, first the pictures I.sub.2, S.sub.2 from bitstream 2 are transmitted and then the following pictures from bitstream 1 are transmitted.

Fast-forward:

[0067] Figure 8 is an illustration of a fast-forward process using SP-pictures. If the bitstream 2 constitutes of only SP-pictures predicted from each other but at larger temporal intervals (e.g. 1 sec) as illustrated, SP-pictures can be used to obtain "Fast Forward" functionality. Furthermore, "Fast Forward" can start and stop at any location in the bitstream. Similarly, "Fast Backward" functionality can be obtained.

Video Redundancy Coding:

[0068] SP-pictures have other uses in applications in which they do not act as replacements of I-pictures. Video Redundancy Coding can be given as an example (VRC). "The principle of the VRC method is to divide the sequence of pictures into two or more threads in such a way that all camera pictures are assigned to one of the threads in a round-robin fashion. Each thread is coded independently. In regular intervals, all threads converge into a so-called sync

frame. From this sync frame, a new thread series is started. If one of these threads is damaged because of a packet loss, the remaining threads stay intact and can be used to predict the next sync frame. It is possible to continue the decoding of the damaged thread, which leads to slight picture degradation, or to stop its decoding which leads to a drop of the frame rate. Sync frames are always predicted out of one of the undamaged threads. This means that the number of transmitted I-pictures can be kept small, because there is no need for complete re-synchronization." For the sync frame, more than one representation (P-picture) is sent, each one using a reference picture from a different thread. Due to the usage of P-pictures these representations are not identical. Therefore, mismatch is introduced when some of the representations cannot be decoded and their counterparts are used when decoding the following threads. Usage of SP-pictures as sync frames eliminates this problem.

Error Resiliency/Recovery:

[0069] Multiple representations of a single frame in the form of SP-frames predicted from different reference pictures, e.g., the immediate previously reconstructed frames and a reconstructed frame further back in time, can be used to increase error resilience. Now, consider the case when an already encoded bitstream is being streamed and there has been a packet loss leading to a frame loss. The client signals the lost frame(s) to the sender which responds by sending the next SP-frame in the representation that uses frames that have been already received by the client.

[0070] We have described the application of SP-pictures in different application/functionality scenarios. Note that the bitstreams in the applications discussed above could have different bitrates, frame sizes and frame rates. Depending on the client's available bandwidth, decoding and viewing capabilities, the appropriate streams can be streamed and moreover, the streams could be dynamically changed to accommodate any changes in these. In the following, we provide a detailed description of SP-picture encoding/decoding within the context of H.26L.

[0071] SP-frame comprises blocks encoded using only spatial correlation among the pixels (intra blocks) and blocks encoded using both spatial and temporal correlation (inter blocks).

[0072] For each inter block:

- The prediction of this block, P(x,y), is formed using received motion vectors and a reference frame.
- The transform coefficients C.sub.pred for P(x,y) corresponding to basis functions f.sub.ij(xy) are calculated.
- The quantized values of c.sub.pred are denoted as l.sub.pred and the dequantized values of l.sub.pred are denoted as d.sub.pred.
- Quantized coefficients l.sub.err, for the prediction error are received from the decoder. The dequantized values of these coefficients will be denoted as d.sub.err.

[0073] Value of each pixel S(x,y) in the inter block is decoded as a weighted sum of the basis functions f.sub.ij(x,y) where the weigh values d.sub.rec will be called dequantized reconstruction image coefficients. The values of d.sub.rec, have to be such that coefficients c.sub.rec, exist by which quantization and dequantization d.sub.rec can be obtained. In addition, values d.rec have to fulfill one of the following conditions:

d.sub.rec = d.sub.pred + d.sub.err; or

C.sub.rec = C.sub.pred + d.sub.err.

[0074] Values S(x,y) can be further normalized and filtered.

[0075] How to utilize SP-frames to switch between different bitstreams is explained in Figure 5. Within each encoded bitstream, SP-pictures should be placed at locations at which one wants to allow switching from one bitstrearm to another (pictures S.sub.1 (513), and S.sub.2 (523) in Figure 5). When

switching from bitstream 1 (510) to bitstream 2 (520), another picture of this type will be transmitted (Figure 5 picture S.sub.12 (550) will be transmitted instead of S.sub.2 (523)). Pictures S.sub.2 (523) and S.sub.12 (550) in Figure 5 are represented by different bitstreams. However, their reconstructed values are identical.

[0076] The invention is described in view of certain embodiments. Variations and modification are deemed to be within the spirit and scope of the invention. The following describes the preferred implementation of the invention as illustrated in Figure 6.

[0077] In the preferred mode of implementation coefficients d.sub.rec are calculated as follows:

- Form prediction of current block, P(x,y), using received motion vectors and the reference frame.
- Calculate transform coefficients c.sub.pred for P(x,y) corresponding to basis functions f.sub.ij(x,y) (Transform block 660). Quantize these coefficients (Quantization block 670). The quantized values will be referred to as quantized prediction image coefficients and denoted as l.sub.pred.
- Obtain quantized reconstruction image coefficients l.sub.rec by adding the received quantized coefficients for the prediction error l.sub.err to l.sub.pred, i.e., l.sub.rec = l.sub.pred - l.sub.err.
- Dequantise l.sub.rec. The dequantised coefficients, output of the Inverse Quantization block, are equal to d.sub.rec.

[0078] In the following, we describe the encoding of SP-frames for the decoder structure described as the preferred embodiment of the invention.

[0079] As can be observed from Figure 5, there are two types of SP-frames, specifically, the SP-frames; placed within the bistream, e.g., S.sub.1 (513) and S.sub.2 (523) in Figure 5, and the SP-frames (S.sub.12 in Figure 5) that will be

sent when there is a switch between bitstreams (from bitstrearn 1 to bitstream 2). The encodings of S.sub.2 (523) and S.sub.12 (550) are such that their reconstructed frames are identical although they use different reference frames as described below.

[0080] First, we describe the encoding of SP-frames placed within the bitstream, e.g., S.sub.1 (513) and S.sub.2 (523) in Figure 5. The original frame is partitioned into blocks and each inter coded block is predicted from one of the earlier reconstructed frames. The prediction frame is formed as described in above. The transform coefficients c.sub.orig and c.sub.pred are calculated for the original frame I(x,y) and prediction frame P(x,y), respectively. C.sub.orig and C.sub.pred are quantized to obtain I.sub.orig and I.sub.pred, respectively. The quantized prediction error coefficients I are then obtained by subtracting I.sub.orig from I.sub.pred, i.e., I.sub.err,= I.sub.orig - I.sub.pred. Motion vectors and the quantized prediction error coefficients are encoded using VLC and the corresponding bitstream is transmitted to the decoder.

[0081] Let I^2.sub.err and I^2.sub.pred denote the quantized coefficients of the prediction error and the prediction frame, respectively, obtained from encoding of S.sub.2. With the procedure described above. Note that in this case, quantized reconstruction image coefficients are given by I^2.sub.rec = I^2.sub.err - I^2.sub.pred. Assume that there will be a switch from bitstream 1 (510) to bitstream 2 (520) at S.sub.2 (523). The encoding of S.sub.12 (550) follows the same procedures as in the encoding of S.sub.2 (523) with the following exceptions: The first difference is that the reference frames are the reconstructed frames obtained from the decoding of the bitstream 1 up to the current frame. Secondly, the quantized prediction error coefficients are calculated as follows: 1^{12}.sub.err = I^2.sub.rec - I^{12}.sub.pred where and I^{12}.pred denotes the quantized prediction image coefficients. The updated quantized prediction error coefficients and the motion vectors are transmitted to the decoder.

[0082] When decoding frame S.sub.12 (550), using the reconstructed frames from bitstream 1 before the switch, coefficients I^{12}.sub.pred are constructed and

added to the received quantized prediction error coefficients I^{12}.sub.err as described above, i.e., I^{12}.sub.rec = I^{12}.sub.err + I^{12}.sub.pred = I^{12}.sub.rec + I^{12}.sub.pred - I^{12}.pred = I^{2}.sub.rec. Note that I^{12}.rec and I^{2}.sub.rec are identical, i.e., S.sub.2 and S.sub.12 are identical. In summary, although S.sub.12 (550) and S.sub.2 (523) have different reference frames, they have identical reconstruction values.

[0083] The changes required in H.26L in order to implement this embodiment of the present invention are described. Although H.26L is used an example standard, embodiments of the present invention and any variations and modifications therefrom are deemed to be within the spirit of scope of the invention.

SP-Picture Decoding

[0084] Additional picture types Ptype Code_number=5 is added to H.26L for signaling SP-picture:

[0085] SP-picture has the same syntax as P-picture. However, interpretation of some of the syntax element differs for Inter and Copy type macroblocks. The macroblocks with type "Inter" are reconstructed as follows:

[0086] 1. Decode levels (both a magnitude and a sign) of the prediction error coefficients, L.sub.err, and motion vectors for the macroblock.

[0087] 2. After motion compensation, for each 4x4 block in the predicted macroblock, perform forward transform.

[0088] An example of a forward transform is provided by Gisle Bjontegaard, "H.26L Test Model Long Term Number 5 (TML-5) draft0", document Q15-K-59, ITU-T Video Coding Experts Group (Question 15) Meeting, Oregon, USA 22-25 August, 2000. Instead of DCT, an integer transform with basically the same coding property as a 4x4 DCT is used. The transformation of some pixels a,b,c,d (say) into 4 transform coefficients A,B,C,D may be defined by:

```
A = 13a + 13b + 13c + 13d
B = 17a +  7b -  7c - 17d
C = 13a - 13b - 13c + 13d
```

```
D =  7a - 17b + 17c -  7d
```

[0089] The inverse transformation of transform coefficients A,B,C,D into 4 pixels a',b',c',d' is defined by:

```
a' = 13A + 17B + 13C +  7D
b' = 13A +  7B - 13C - 17D
c' = 13A -  7B - 13C + 17D
d' = 13A - 17B + 13C -  7D
```

[0090] Due to the fact that the expressions above are not normalized, a' = 676a. Normalization may be performed in the quantization/dequantization process and a final shift after inverse quantization.

[0091] The transform/inverse transform may be performed both vertically and horizontally in the same manner as in H.263.

[0092] For chroma component, an additional 2x2 transform for the DC coefficients may be performed. The 2 dimensional 2x2 transform procedure is illustrated below. DC0,1,2,3 are the DC coefficients of 2x2 chroma blocks.

DC0 DC1 Two dimensional 2x2 transform $\Rightarrow$ DDC(0,0) DDC(1,0)

DC2 DC3 DDC(0,1) DDC(1,1)

Definition of transform:

DCC(0,0) = (DC0+DC1+DC2+DC3)/2

DCC(1,0) = (DC0-DC1+DC2-DC3)/2

DCC(0,1) = (DC0+DC1-DC2-DC3)/2

DCC(1,1) = (DC0-DC1-DC2+DC3)/2

Definition of inverse transform:

DC0 = (DCC(0,0)+ DCC(1,0)+ DCC(0,1)+ DCC(1,1))/2

DC1 = (DCC(0,0)- DCC(1,0)+ DCC(0,1)- DCC(1,1))/2

DC2 = (DCC(0,0)+ DCC(1,0)- DCC(0,1)- DCC(1,1))/2

DC3 = (DCC(0,0)- DCC(1,0)- DCC(0,1)+ DCC(1,1))/2

[0093] Quantize obtained coefficients K L.sub.pred =(KxA(QP) + $0.5x2^{20}$)/2^{20} where A(QP) is defined below in the following example from the art.

[0094] An example of quantizing is provided by Gisle Bjontegaard, "H.26L Test Model Long Term Number 5 (TML-5) draft0", document Q15-K-59, ITU-T Video Coding Experts Group (Question 15) Meeting, Oregon, USA 22-25 August, 2000.

[0095] The quantization/dequantization process may perform 'normal' quantization/dequantization as well as take care of the above transform process which did not contain normalization of transform coefficients. 32 different QP values may be used.

[0096] The QP signaled in the bitstream applies for luma quantization/dequantization referred to as QP.sub.luma. For chroma quantization/dequantization a different value - QP.sub.chroma - is used. The relation between the two is:

QP.sub.luma

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

QP.sub.chroma

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 17 18 19 20 20 21 22 22 23 23 24 24 25 25

[0097] When QP is used in the following we mean QP.sub.luma or QP.sub.chroma depending on what is appropriate.

[0098] Two arrays of numbers are used for quantization/dequantization.

A(QP=0,..,31)

620, 553, 492, 439, 391, 348, 310, 276, 246, 219, 195, 174, 155, 138, 123, 110, 98, 87, 78, 69, 62, 55, 49, 44, 39, 35, 31, 27, 24, 22, 19, 17

B(QP=0,..,31)

3881,4351,4890,5481,6154,6914,7761,8718,9781,10987,12339,13828,15523,17435,19561, 21873,24552,27656,30847,34870,38807,43747,49103,54683,61694,68745,77615,89113,10 0253,109366,126635,141533

[0099] The relation between A() and B() is: $A(QP)xB(QP)x676^2 = 2^{40}$.

[00100] It is assumed that a coefficient K is quantized in the following way:

[00101] LEVEL = $(KxA(QP) + fx2^{20})/2^{20}$ f is in the range (0-0.5) and f has the same sign as K.

[00102] Dequantization:

$$K' = LEVELxB(QP)$$

After inverse transform this results in pixel values that are 2^{20} too high. A shift of 20 bits (with rounding) is therefore needed on the reconstruction side. The definition of transform and quantization is designed so that no overflow will occur with use of 32 bit arithmetic.

[00103] The coefficients of K is obtained by:

$$L.sub.pred = (KxA(QP) + 0.5x2^{20})/2^{20}$$

[00104] Add the quantized prediction image coefficients, L.sub.pred, to the prediction error coefficients levels, i.e., L.sub.rec = L.sub.err + L.sub.pred.

[00105] 3. The coefficients, L.sub.rec, are dequantized and inverse transform is performed for these dequantized levels. Dequantization and inverse transform are performed. The reconstructed values are equal to the result of the inverse transformation shifted by 20 bit (with rounding) as described above.

[00106] For Copy type macroblocks, only steps 2 and 3 are performed. While applying the deblocking filter, both Inter and Copy macroblocks are treated as Intra macroblocks with coefficients represented by L.sub.rec.

SP-Picture Encoding

[00107] SP-picture placed within a single bitstream (pictures S.sub.1 (513 of FIG. 5 and S.sub.2 (523 of FIG. 5). The encoding and decoding of SP-picture which is transmitted when switching from one bitstream to another (picture S.sub.12 (550) and picture S.sub.2 (523)) are described below.

[00108] When encoding an SP-picture placed within a bitstream, the prediction error coefficients for luminance may be obtained as follows:

[00109] 1. After motion compensation, for each 4x4 block in the predicted macroblock and in the original image, forward transform is performed. For chroma component an additional 2x2 transform for DC coefficients is performed. The transform coefficients for the original image are denoted as K.sub.orig and for the predicted image as K.sub.pred.

[00110] 2. Transform coefficients for the predicted blocks are quantized . Obtained levels are denoted as L.sub.pred.

[00111] 3. The prediction error coefficients are obtained by K.sub.err = K.sub.orig - L.sub.pred $x2^{20}/A(QP)$ and can be quantized.

[00112] Let as assume that we want to encode the SP-picture, denoted as S.sub.12, to switch from bitstream 1 to bitstream 2. The reconstructed values of this picture have to be identical to the reconstructed values of SP-picture in bitstream 2, denoted as S.sub.2, to which we are switching. The bitstream of the Intra macroblocks in frame S.sub.2 are copied to S.sub.12. The encoding of Inter macroblocks is performed as follows:

[00113] 1. Form the predicted frame for S.sub.12 by performing motion estimation with the reference frames being pictures preceding S.sub.1 in bitstream 1.

[00114] 2. Perform for each 4x4 block in the predicted macroblock 4x4 forward transform. An additional 2x2 transform for DC coefficients of the chroma component is performed.

[00115] 3. Quantize the obtained coefficients and subtract the quantized coefficient levels from the corresponding L.sub.rec of S.sub.2-picture. The resulting levels are the levels of the prediction error which will be transmitted to the decoder.

[00116] Figure 9 illustrates a comparison of the coding efficiency of each picture type, namely I, P and SP frames in terms of their PSNR as a function of bitrate performances for the selected sequences (container and hall sequences). These results are generated by encoding every frame in the sequence with the

same picture type, i.e., I, P or SP, except the first frame which is always an I-frame. As can be observed from Figure 8, the coding efficiency of an SP-picture is worse than P-frames while it is significantly much better than that of I-frames. Although the coding efficiency of each picture type is important, it is important to note that the SP-frames provide functionalities that are usually achieved only with I-frames.

[00117] In the following, we illustrate the simulation results when SP and I-frames are introduced at fixed intervals. Figure 10 illustrates the results obtained with the following conditions: The first frame is encoded as an I-Picture and at fixed intervals, in this case 1sec, the frames are encoded as I or SP–pictures while the rest of the frames are encoded as P-pictures.

[00118] Also included in Figure 10 is the performance achieved when all the frames are encoded using P-frames. Note that in this case, none of the functionalities mentioned earlier can be obtained while this provides a benchmark for comparison of both SP and I-picture cases. As can be observed from Figure 10, SP-pictures, while providing the same functionalities as an I-picture, yield significantly better performance in terms of PSNR as a function of bitrate. For example for the Hall sequence around 40Kbps, there is 2-2.5dB improvement when SP-frames are used instead of I-frames while there is 0.5dB penalty over the benchmark all P-frame conditions.

[00119] Figure 11 demonstrates the performance improvement using SP-pictures instead of I-frames for "Fast-forward". We also include the performance achieved by using only P-frames. Note again, in this case, restarting playing is not possible without a mismatch. Nevertheless, this provides another benchmark for comparison of the other schemes. As can be observed from Figure 11, there is significant improvement with SP-pictures over I-pictures. For container sequence at 10Kbps, an improvement of 5.5dB can be obtained.

[00120] In another embodiment of the present invention, the coding efficiency of SP-frames is improved by using a separate quantization value for the predicted frame than the prediction error coefficients. The changes required in

H.26L in order to implement this embodiment of the present invention are described. Although H.26L is used an example standard, embodiments of the present invention and any variations and modifications therefrom are deemed to be within the spirit of scope of the invention.

Another embodiment for SP-Picture decoding

[00121] Picture types Ptype Code_number=5 is added to H26.L standard to signal SP-picture. If the Ptype indicates SP-picture, an additional codeword SPQP(5 bits) follows PQP codeword. Otherwise SP-picture has the same syntax as P-picture; however, interpretation of some of the syntax element may differ for Inter and Copy type macroblocks.

[00122] Additional array of numbers is used when decoding SP-picture:

$$F(QP)=2^{20} / A(QP)$$

where constant A(QP) is defined above in the section on quantization.

[00123] Decoding of the luma component is described first. The macroblocks with type "Inter" are reconstructed as follows:

[00124] 1. Decode levels (both a magnitude and a sign) of the prediction error coefficients, L.sub.err, and motion vectors for the macroblock.

[00125] 2. After motion compensation, for each 4x4 block in the predicted macroblock, perform forward transform as described above. Using the resulting prediction image coefficients K.sub.pred and prediction error coefficient levels L.sub.err calculate coefficients

$$K.sub.rec = (K.sub.pred + L.sub.err \times F(QP.sub.1))$$

and quantize them:

$$L.sub.rec = (K.sub.rec \times A(QP.sub.2) + 0.5 \times 2^{20}) / 2^{20}.$$

[00126] The value of QP.sub.1 is given by PQP and QP.sub.2 by SPQP.

[00127] 3. The coefficients, L.sub.rec, are dequantized using QP.sub.2 and the inverse transform is performed for these dequantized levels. Dequantization and inverse transform are performed as described in Gisle Bjontegaard, "H.26L Test Model Long Term Number 5 (TML-5) draft0", document Q15-K-59, ITU-T Video Coding Experts Group (Question 15) Meeting, Oregon, USA 22-25 August, 2000. The reconstructed values are equal to the result of the inverse transformation shifted by 20 bit (with rounding).

[00128] For Copy type macroblocks, only the steps 2 and 3 are performed. While applying deblocking filter, both Inter and Copy macroblocks are treated as Intra macroblocks with coefficients represented by L.sub.rec.

[00129] Dequantization of the chroma component is performed in a similar manner with the following differences. In step 2 additional 2x2 transform for the DC coefficients is performed after 4x4 transform. Values of QP.sub.1 and QP.sub.2 are changed according to the relation between QP values used for luma and chroma specified above.

Another embodiment for SP-Picture Encoding

[00130] When encoding an SP-picture placed within a bitstream, the prediction error coefficients for luminance can be obtained as follows:

[00131] 1. After motion compensation, for each 4x4 block in the predicted macroblock and in the original image, forward transform is performed. The transform coefficients for the original image are denoted as K.sub.orig and for the predicted image as K.sub.pred.

[00132] 2. Transform coefficients for the predicted blocks are quantized using QP=QP.sub.2 as specified above with f=0.5. The resulting levels are denoted as L.sub.pred.

[00133] 3. The prediction error coefficients K.sub.err = K.sub.orig - L.sub.pred x F(QP.sub.2) can be quantized using one of the methods described in G. Bjontegaard, "H.26L Test Model Long Term Number 6 (TML-6) draft0", document VCEG-L45, ITU-T Video Coding Experts Group Meeting, Eibsee, Germany, 09-12 January 2001, with QP= QP.sub.1.

[00134] The same procedure is used to calculate coefficients for chrominance with the following differences : In step 1, an additional 2x2 transform for DC coefficients is performed after 4x4 transform. Values of QP_1 and QP_2 are changed according to the relation between QP values used for luma and chroma specified in above.

[00135] Let as assume that we want to encode the SP-picture, denoted as S.sub.12, to switch from bitstream 1 (510) to bitstream 2 (520) in FIG. 5. The reconstructed values of this picture have to be identical to the reconstructed values of SP-picture in bitstream 2 (520), denoted as S.sub.2 (523), to which we are switching. The bitstream of the Intra macroblocks in frame S.sub.2 (523) are copied to S.sub.12 (550). The encoding of Inter macroblocks is performed as follows:

[00136] 1. Form the predicted frame for S.sub.12 by performing motion estimation with the reference frames being pictures proceeding S.sub.1 (513) in bitstream 1 (510).

[00137] 2. Perform for each 4x4 block in the predicted macroblock 4x4 forward transform followed by an additional 2x2 transform for DC coefficients for chroma component.

[00138] 3. Quantise obtained coefficients and subtract the quantized coefficient levels from the corresponding levels L.rec of S.sub.2 picture. Use QP equal to value specified by SPQP codeword of frame S.sub.2. The resulting levels are the levels of the prediction error which will be transmitted to the decoder through a communication system. Notice that for frame S.sub.12 QP values specified by PQP and SPQP are same and equal to the value given by SPQP codeword of frame S.sub.2.

[00139] A novel SP-picture encoding/decoding method which uses different quantization values for the predicted block and the prediction error coefficients has been provided. The use of two different values of QP allows to trade off between coding efficiency of SP-pictures placed within a single bitstream and SP-pictures used when switching from one bitstream to another. The lower the

value of SPQP with respect to PQP, the higher the coding efficiency of SP-picture placed within a single bitstream, while, on the other hand, the larger number of bits is required when switching to this picture. The choice of the SPQP value can be application dependent. For example, when SP-pictures are used to facilitate random access one can expect that SP frames placed within a single bitstream will have the major influence on compression efficiency and therefore SPQP value should be small. On the other hand, when SP pictures are used for streaming rate control, SPQP value should be kept close to PQP since SP-pictures sent during switching from one bitsream to another will have large share of the overall bandwidth.

[00140] While the preferred embodiment and various alternative embodiments of the invention has been disclosed and described in detail herein, it will be obvious to those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope thereof.